

# Python

# О языке

Python — современный язык программирования, работающий на всех распространенных операционных системах для настольных компьютеров.

Python — современный универсальный интерпретируемый язык программирования.

Python — это язык с динамической типизацией

# Достоинства

1. Кроссплатформенность и бесплатность.
2. Простой синтаксис и богатые возможности позволяют записывать программы очень кратко, но в то же время понятно.
3. По простоте освоения язык сравним с бейсиком, но куда более богат возможностями и значительно более современен.
4. Богатая стандартная библиотека, возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т.д.)

# Синтаксис

Синтаксис языка Python, как и сам язык, очень прост. Он не содержит сложных неинтуитивных конструкций, поэтому достаточно прост в изучении.

Конец строки является концом инструкции (точка с запятой не требуется).

```
a = 5  
b = 3  
print(a + b)
```

Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков.

```
if a == 5:  
    print('yes')  
    a += 1
```

Built-in слова и идентификаторы

# Идентификаторы

Идентификатор — имя некоторого объекта в программе, являющееся его уникальным признаком, позволяющим отличать его от других объектов

Идентификатор обязательно есть у каждой переменной, функции, объекта и т.п. Идентификаторы в Питоне не ограничены по длине и чувствительны к регистру. То есть `A` и `a` — это разные имена переменных или функций.

# Имена переменных и ключевые слова

## Хорошо

summ  
summ\_of\_variables  
\_new\_sum  
is\_payable  
\_\_another\_variable

## нормально

Summ  
summOfVariables  
New\_sum  
isPayable  
\_\_anotherVariable

## Нельзя

Sum%m  
\$summOfVariable\$  
42New\_sum  
isP\*yable  
\_\_anotherV@riable



# Ключевые слова

<code>False</code>	<code>class</code>	<code>finally</code>	<code>is</code>	<code>return</code>
<code>None</code>	<code>continue</code>	<code>for</code>	<code>lambda</code>	<code>try</code>
<code>True</code>	<code>def</code>	<code>from</code>	<code>nonlocal</code>	<code>while</code>
<code>and</code>	<code>del</code>	<code>global</code>	<code>not</code>	<code>with</code>
<code>as</code>	<code>elif</code>	<code>if</code>	<code>or</code>	<code>yield</code>
<code>assert</code>	<code>else</code>	<code>import</code>	<code>pass</code>	
<code>break</code>	<code>except</code>	<code>in</code>	<code>raise</code>	

# Виртуальное окружение

Главная задача виртуальной среды Python – создание изолированной среды для проектов Python.

Каждый проект может иметь свои собственные зависимости, вне зависимости от того, какие зависимости у другого проекта.

# Комментарии

**Комментарии** – это строки, которые существуют в коде программы, но игнорируются компиляторами и интерпретаторами.

**Комментарии** делают код более удобочитаемым, так как позволяют предоставить пользователям дополнительную информацию или добавить объяснение того или иного блока кода

## Хорошие против плохих комментариев

это довольно бесполезный комментарий:

```
b = 56  # assigning b a value of 56
```

Типы

# Типы комментариев

## Однострочные комментарии

```
# defining the post code
```

```
code = 75000
```

```
# define the general structure of the product with default values
```

```
product = {
```

```
    "productId": 0,          # product reference id, default: 0
```

```
    "description": "",       # item description, default: empty
```

```
    "categoryId": 0,        # item category, default: 0
```

```
    "price": 0.00           # price, default: 0.00
```

```
}
```

## Многострочные комментарии

```
# LinuxThingy version 1.6.5
#
# Parameters:
#
# -t (--text): show the text interface
# -h (--help): display this help
```

```
"""
```

```
LinuxThingy version 1.6.5
Parameters:
-t (--text): show the text interface
-h (--help): display this help
```

```
"""
```

## Комментарии документации

```
def add(value1, value2):  
    """Calculate the sum of value1 and value2."""  
    return value1 + value2
```

# Ввод входных данных

```
>>> string = input('Input some lint\n')
```

Input some lint

line

```
>>> string
```

```
'line'
```



# Вывод входных данных

```
>>> string = input('Input some lint\n')
```

Input some lint

```
>>> print(string)
```

line











